

图文 75 生产案例：从 RocketMQ 底层原理分析为什么会重复发优惠券？

433 人次阅读 2020-01-09 07:00:00

详情 评论



狸猫技术

进店逛

相关频道



从 0 开
件实站
已更新9

生产案例：从 RocketMQ 底层原理分析为什么会重复发优惠券？



继《从零开始带你成为JVM实战高手》后，救火队长携新作再度出山，重磅推荐：

(点击下方蓝字试听)

[《从零开始带你成为MySQL实战优化高手》](#)

1、客服反馈的奇怪问题：有用户重复收到了多个优惠券

话说小猛的团队前段时间对MQ的消息丢失问题进行了系统的研究，同时对系统的核心流程引入了一套消息零丢失的方案，保证系统的核心消息数据不会丢失，上线之后效果非常好，再也没出现过用户支付后没发红包的问题了。

但是今天客服突然又给技术团队反馈了一个问题，说是有用户在支付一个订单之后，一下子收到了多个优惠券，本来按照规则只应该有一个优惠券的！

也就是说，我们给用户重复发放了一个优惠券！

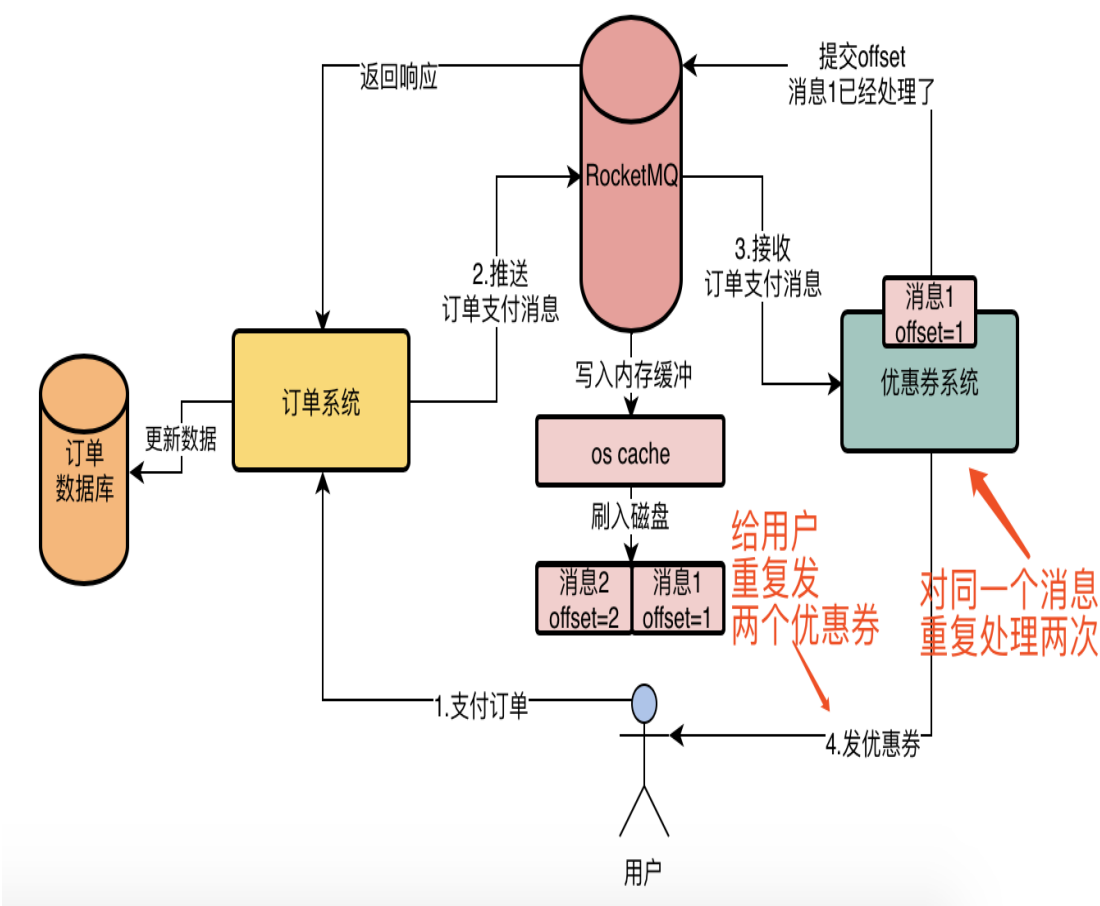
于是针对这个问题，小猛的团队又展开了一番研究。

2、问题的定位：优惠券系统重复消费了一条消息

从之前介绍的项目背景中，我们都可以知道一点，那就是订单系统已经跟各个系统进行了解耦，也就是说当订单支付成功之后，会发送一条消息到MQ里去，然后红包系统从里面获取消息派发红包，优惠券系统从里面获取消息派发优惠券，其他系统也是同理。

所以当小猛的团队对线上系统的日志进行分析的时候，发现了一个奇怪的问题，那就是优惠券系统似乎对同一条消息重复的处理了两次，导致他给一个用户重复的派发了两个优惠券

我们看下图的示意



优惠券重复派发两次的问题已经定位到了，就是优惠券系统对同一个订单支付成功的消息处理了两次，就导致给用户重复发放了两张优惠券。

那么接下来我们需要思考的问题就是，为什么优惠券会对同一个消息重复处理两次呢？

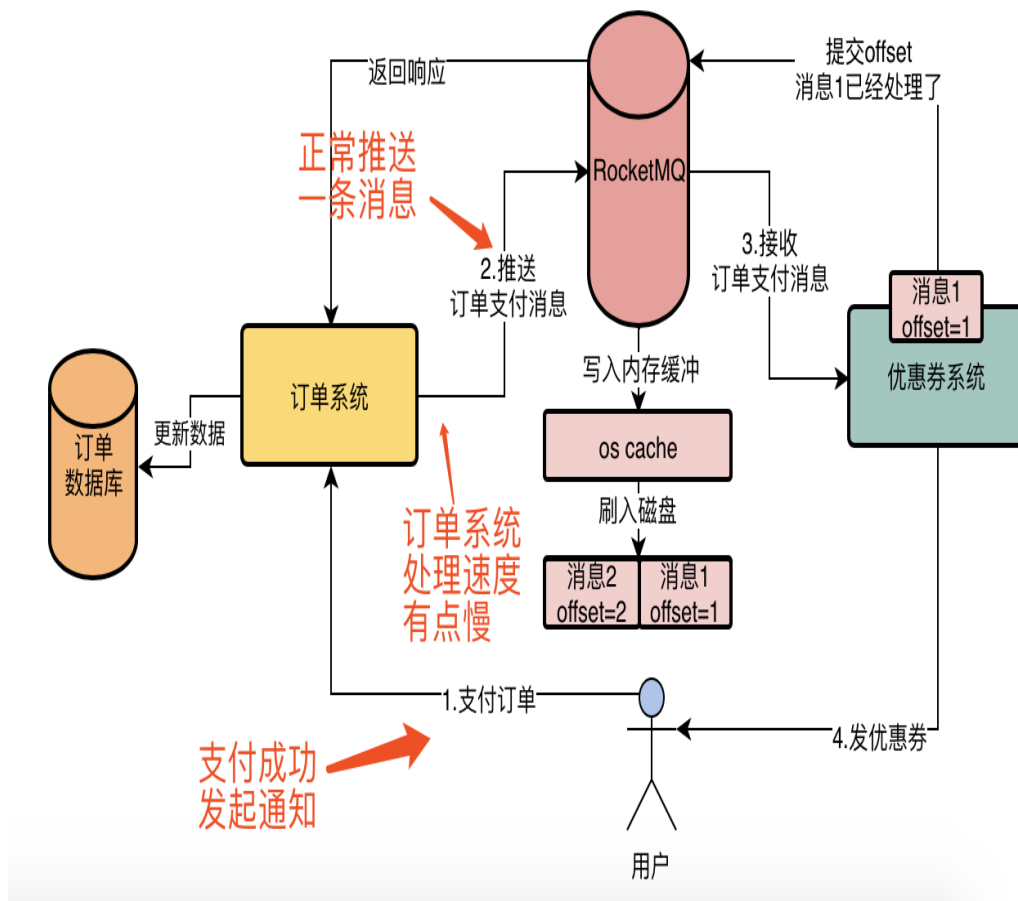
3、订单系统发送消息到MQ的时候会重复吗？

要搞明白为什么优惠券系统对同一个消息重复处理了两次，我们先来研究第一个问题，我们的订单系统收到一个支付成功的通知之后，他在发送消息到MQ的时候，会重发把一条消息发送两次吗？

可能有的朋友乍一看觉得应该不可能，但是其实在生产环境中运行的系统，显然是有可能把一个消息重复发两次的！

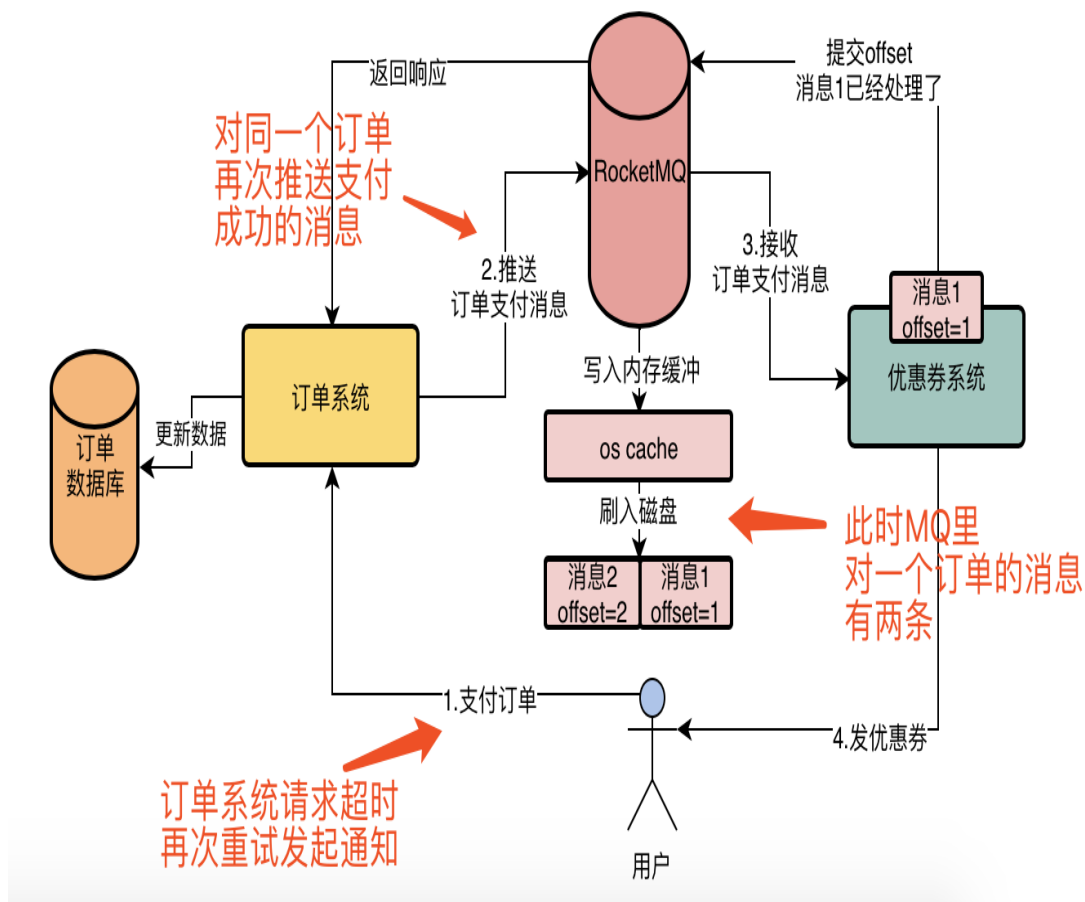
首先，假设用户在支付成功之后，我们的订单系统收到了一个支付成功的通知，接着他就向MQ发送了一条订单支付成功的消息，这个大家都知道没有什么问题。

但是偏偏可能因为不知道什么原因，你的订单系统处理的速度有点慢，我们看下图。

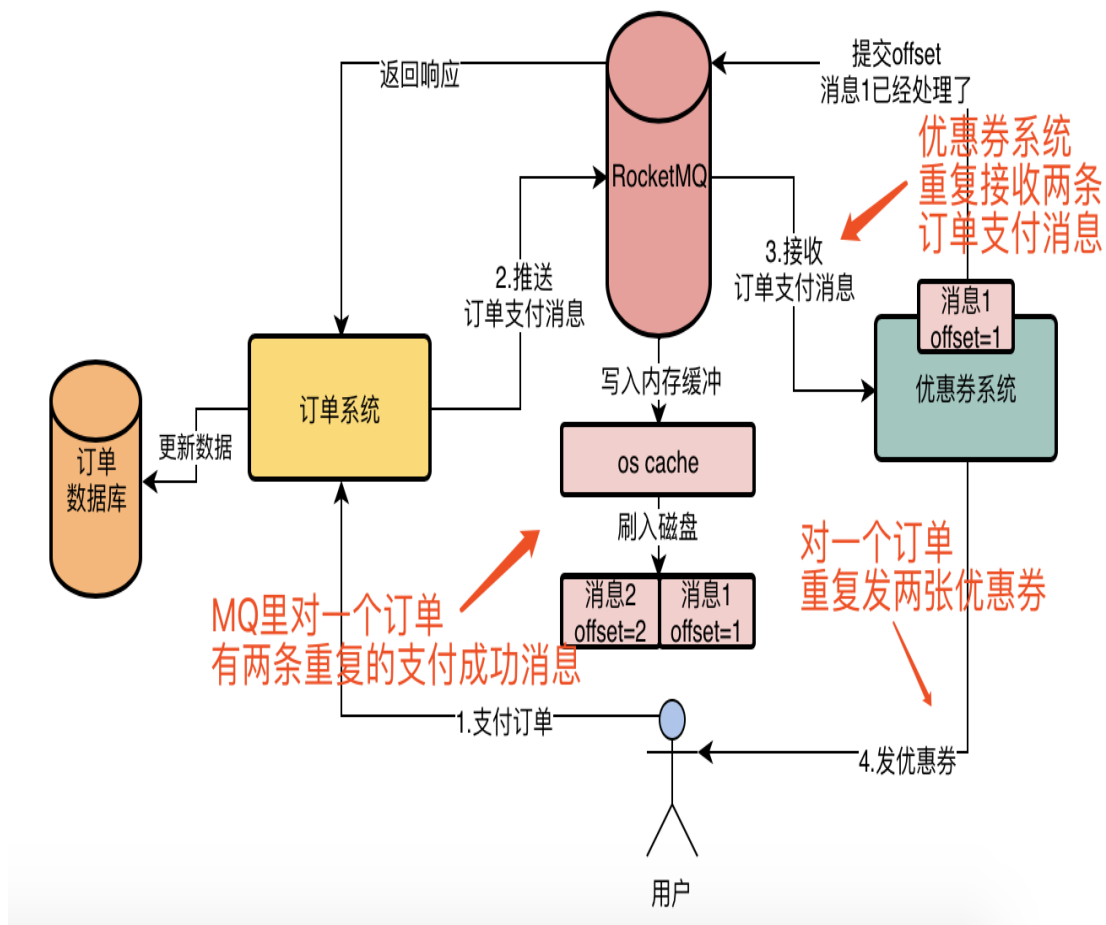


然后可能就因为你的订单系统处理的速度有点慢了，这就导致支付系统跟你订单系统之间的请求出现了超时，此时有可能支付系统再次重试调用了你订单系统的接口去通知你，这个订单支付成功了，然后你的订单系统这个时候可能又一次推送了一条消息到MQ里去，相当于是一个订单支付成功的消息，你重复推送了两次到MQ！

此时相当于是在MQ里就会对一个订单的支付成功消息，总共有两条，我们看下图的示意。



那如果你订单系统对一个订单重复推送了两次支付成功消息到MQ，MQ里对一个订单有两条重复的支付成功消息，优惠券系统必然会消费到一个订单的两条重复的支付成功消息，也必然会针对这个订单给用户重复的派发两个优惠券，我们看下图。



所以大家看到这里，通过一步一图的方式，可以很清晰的看到我们用于发送消息到MQ的订单系统，如果出现了接口超时等问题，可能会导致上游的支付系统重试调用订单系统的接口，进而导致订单系统对一个消息重复发送两条到MQ里去！

4、重试是一把双刃剑：订单系统自己重复发送消息

接着我们来考虑第二种情况，假设支付系统没有对一个订单重复调用你的订单系统的接口，而是你订单系统自己可能就重复发送消息到MQ里去

那这是一个什么情况呢？我们来分析一下。

假设我们的订单系统为了保证消息一定能投递到MQ里去，因此采用了重试的代码，我们之前也讲过这个伪代码的示意

我们看下面的代码片段，如果发现MQ发送有异常，则会进行几次重试。

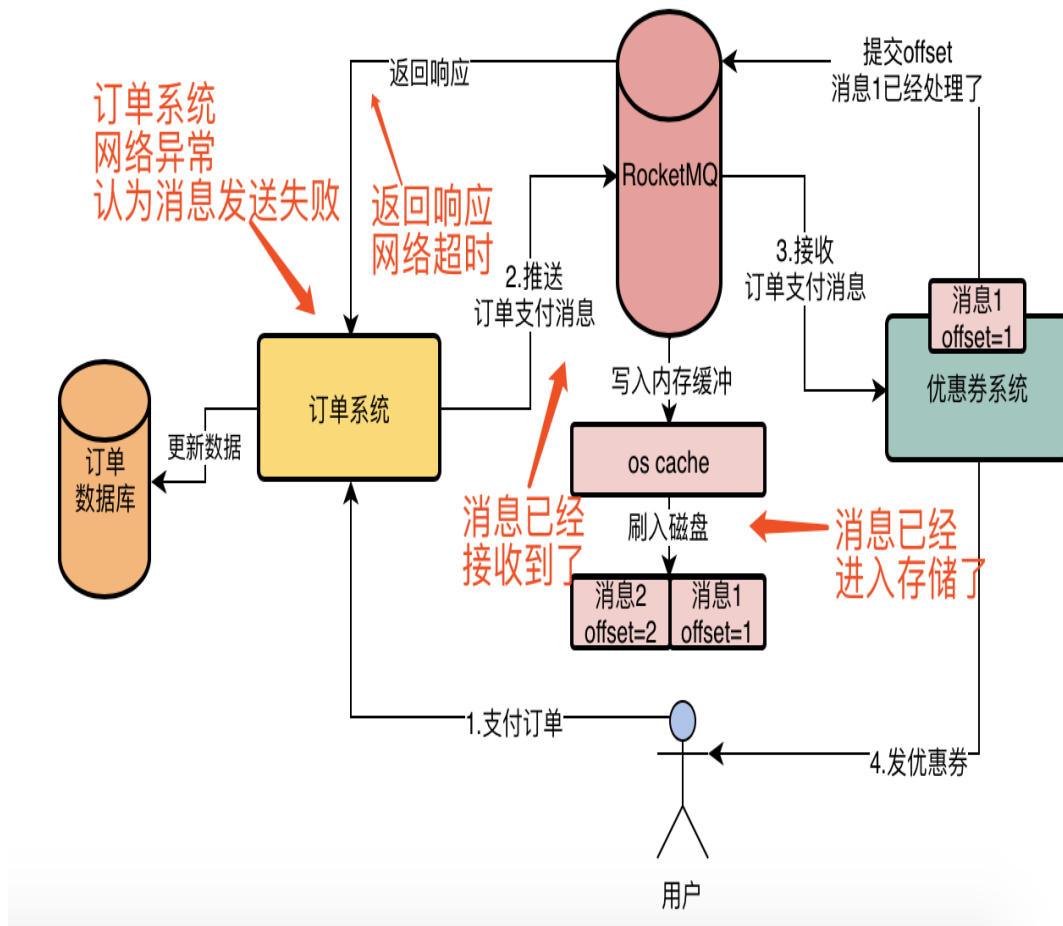
```
try {  
    // 执行订单本地事务  
    orderService.finishOrderPay();  
    // 发送消息到MQ去  
    producer.sendMessage();  
} catch (Exception e) {  
    // 如果发送消息失败了，进行重试  
    for (int i = 0; i < 3; i++) {  
        // 重试发送消息  
    }  
    // 如果多次重试发送消息之后，还是不行  
    // 回滚本地订单事务  
    orderService.rollbackOrderPay();  
}
```

但是这种重试的方式，其实是一把双刃剑，因为正是这个重试就可能导致消息重复发送

我们来考虑一个情况，假设你发送了一条消息到MQ了，其实MQ是已经接收到这条消息了，结果MQ返回响应给你的时候，网络有问题超时了，就是你没能及时收到MQ返回给你的响应。

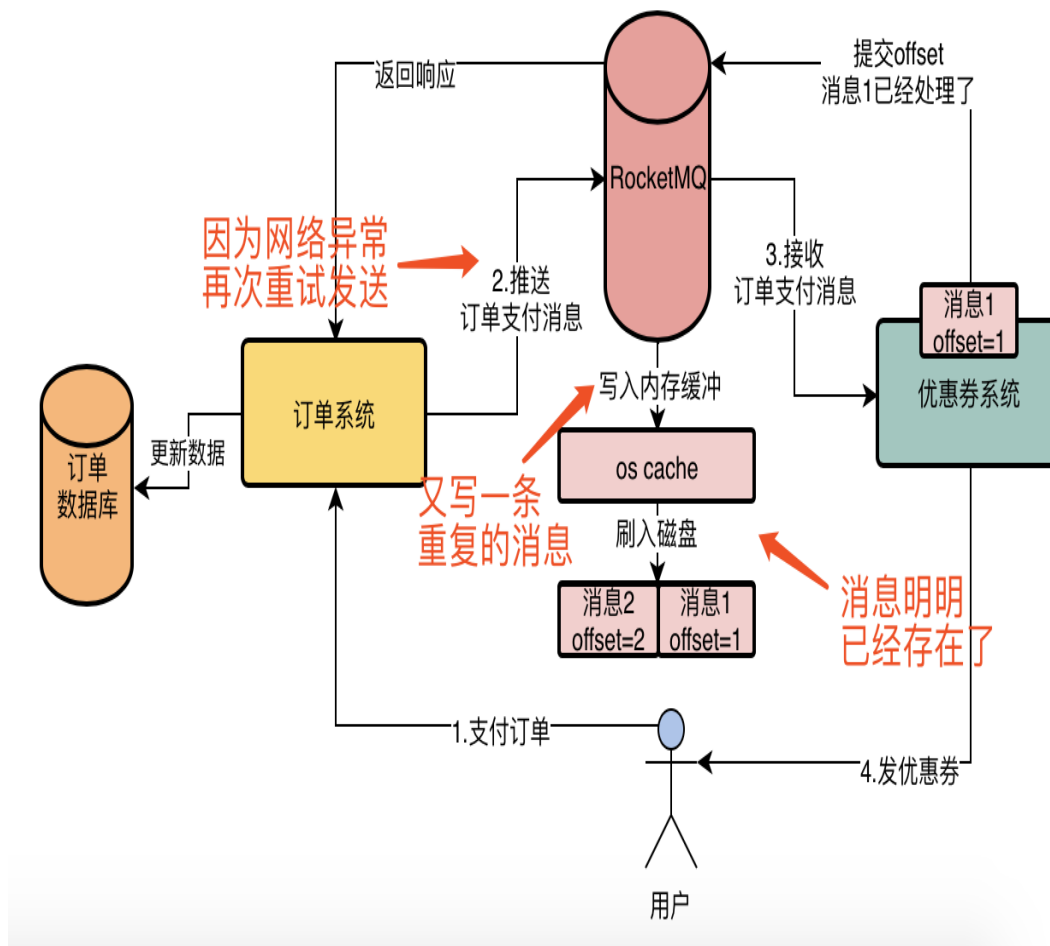
但是大家一定要明确一点，此时MQ里其实是已经有你发送过去的消息了，只不过他返回给你的响应没能给你而已！

我们看下面的图示



这个时候，你的代码里可能会发现一个网络超时的异常，然后你就会进行重试再次发送这个消息到MQ去，然后MQ必然会收到一条一模一样的消息，进而导致你的消息重复发送了！

大家看下图的示意



所以这种重试代码大家在使用的时候一定要小心！因为他还是有一定的概率会导致你重发消息的！

5、优惠券系统重复消费一条消息

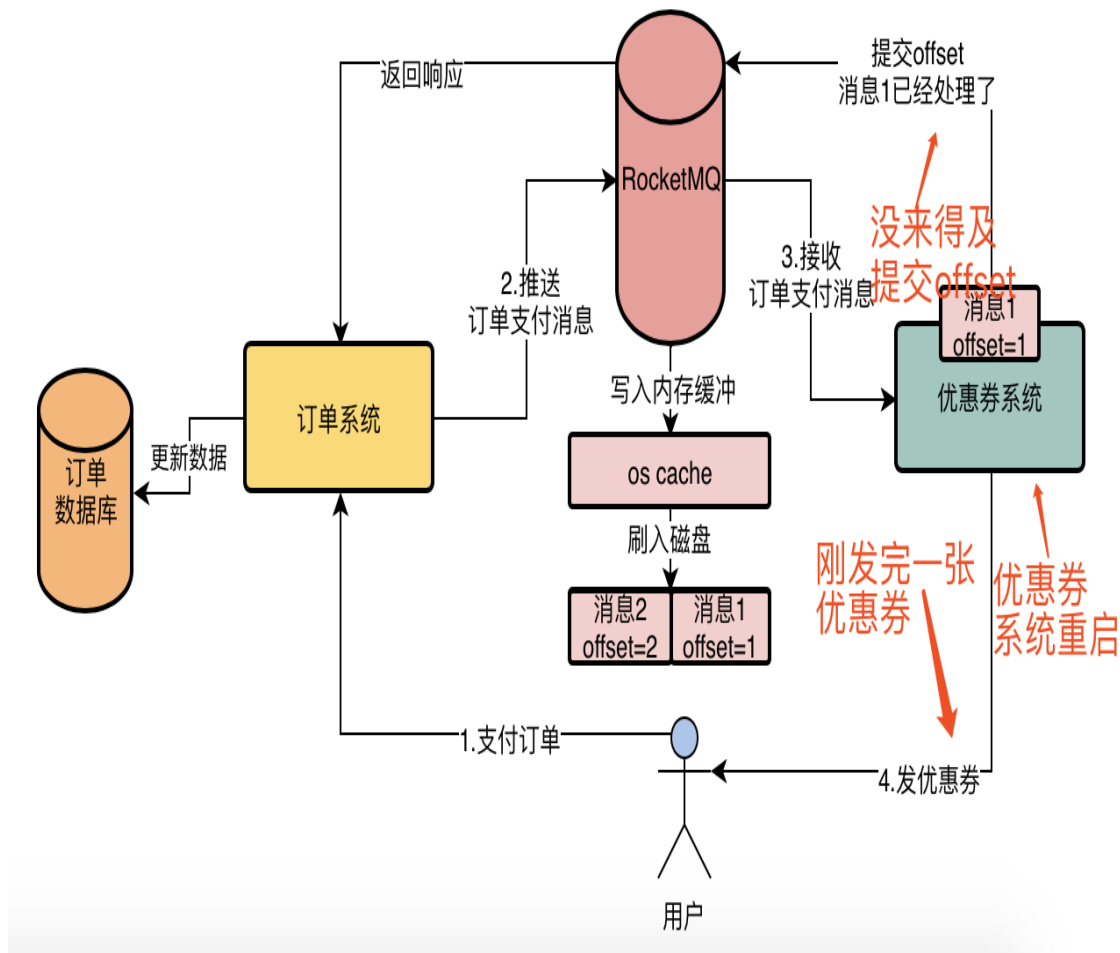
接着我们继续来看，即使你没有重复发送消息到MQ，哪怕MQ里就一条消息，优惠券系统也有可能会重复进行消费

这是为什么呢？我们一步一步来分析一下。

假设你的优惠券系统拿到了一条订单成功支付的消息，然后都已经进行处理了，也就是说都已经对这个订单给你发了一张优惠券了，本来我们之前讲过，这个时候他应该返回一个CONSUME_SUCCESS的状态，然后提交消费进度offset到broker的。

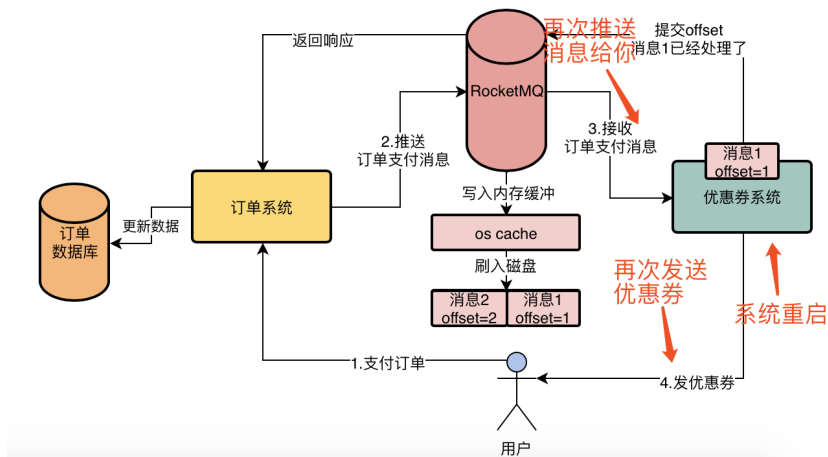
但是不巧的是，你刚刚发完优惠券，还没来得及提交消息offset到broker呢！优惠券系统就进行了一次重启！比如可能优惠券系统的代码更新了，需要重启进行重新部署。

我们看下面的图示



这时因为你没提交这条消息的offset给broker，broker并不知道你已经处理完了这条消息，然后优惠券系统重启之后，broker就会再次把这条消息交给你，让你再一次进行处理，然后你会再一次发送一张优惠券，导致重复发送了两次优惠券！

这就是对同一条消息，优惠券系统重复处理两次的原因，我们看下面的图示。



6、消息重复问题应该是一种家常便饭

实际上大家要知道，对类似优惠券系统这样的业务系统，我们肯定是要频繁的更新代码的，可能每隔几天就需要重启一次系统进行代码的更新

所以其实你重启优惠券系统的时候，可能有一批消息刚处理完，还没来得及提交offset给broker呢，然后你重启之后就会再一次重复处理这批消息，这种情况可能是家常便饭！

另外就是对于系统之间的调用，有的时候出现超时和重试的情况也是很常见的，所以你负责发消息到MQ的系统，很可能时不时的出现一次超时，然后被别人重试调用你的接口，你可能会重复发送一条消息到MQ里去，这可能也是家常便饭！

因此在使用MQ的时候，大家应该对消息重复问题习惯他，当做必须处理的一个问题。

7、小作业：你的系统里出现过消息重复问题吗？

今天给大家留一个小的思考题，希望每个朋友都思考一下：

自己负责的系统中如果使用MQ的话，是否出现过消息重复的问题吗？

如果有的话，是一个什么场景？

对你们有什么影响？

你们是如何发现的？然后是如何解决的？

大家可以踊跃的把自己的经验分享到专栏的评论区里去。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐：

[《从零开始带你成为JVM实战高手》](#)

[《21天互联网Java进阶面试训练营》（分布式篇）](#)

[《互联网Java工程师面试突击》（第1季）](#)

[《互联网Java工程师面试突击》（第3季）](#)

重要说明：

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《付费用户如何加群》（**购买后可见**）